Paper No. 18

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte PAUL STUART WILLIAMSON,
DAVID MICHAEL BENDER,
and MICHAEL JOSEPH REYNOLDS

Appeal No. 2002-2073
Application No. 09/210,104

ON BRIEF

Before FLEMING, LEVY, and BLANKENSHIP, Administrative Patent Judges.

LEVY, Administrative Patent Judge.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134 from the examiner's final rejection of claims 1-24, which are all of the claims pending in this application.

BACKGROUND

Appellants' invention relates to a method and apparatus for monitoring the execution of a program. An understanding of the invention can be derived from a reading of exemplary claim 1, which is reproduced as follows:

1. An Execution Trace Facility computer system, which provides all available pertinent data required to trace a program flow for malfunctions, comprising:

means for inserting said Execution Trace Facility into the code of the program in a disabled mode, at various strategic locations;

means for recompiling said program after said Execution Trace Facility is inserted into said code of the said program;

means for using said program in a normal operating environment and for determining if a malfunction of said program occurs; and

means for enabling said Execution Trace Facility to restart said program and detect and correct said malfunction only in response to a detection that a malfunction of said program has occurred.

The prior art references of record relied upon by the examiner in rejecting the appealed claims are:

Borchardt et al. (Borchardt)	5,513,317	Apr. 30, 1996
Mann	6,094,729	Jul. 25, 2000 (filed Dec. 17, 1997)
Alexander, III et al. (Alexander)	6,118,940	Sep. 12, 2000 (filed Nov. 25, 1997)

Claims 1-3, 6-11, 14-19, and 22-24 stand rejected under 35 U.S.C. § 103 as being unpatentable over Borchardt in view of Mann.

Claims 4, 5, 12, 13, 20, and 21 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Borchardt in view of Mann, and further in view of Alexandria.

Rather than reiterate the conflicting viewpoints advanced by the examiner and appellants regarding the above-noted rejections, we make reference to the examiner's answer (Paper No. 13, mailed March 22, 2002) and the final rejection (Paper No. 6, mailed July 2, 2001) for the examiner's complete reasoning in support of the rejections, and to appellants' brief (Paper No. 12, filed January 23, 2002) and reply brief (Paper No. 14, filed May 7, 2002) for appellants' arguments thereagainst. Only those arguments actually made by appellants have been considered in this decision. Arguments which appellants could have made but chose not to make in the brief have not been considered. See 37 CFR 1.192(a).

OPINION

In reaching our decision in this appeal, we have carefully considered the subject matter on appeal, the rejections advanced by the examiner, and the evidence of obviousness relied upon by the examiner as support for the rejections. We have, likewise, reviewed and taken into consideration, in reaching our decision, appellants' arguments set forth in the briefs along with the examiner's rationale in support of the rejections and arguments in rebuttal set forth in the examiner's answer.

We observe at the outset that appellants assert (brief, page 4) that "claims 1-24 stand or fall together as a single group." Consistent with this statement, appellants arguments are generic to each of the claims. Accordingly, we select claim 1 as representative of claims 1-3, 6-11, 14-19, and 22-24, rejected under 35 U.S.C. § 103(a) as unpatentable over Borchardt in view of Mann, and select claim 4 as representative of claims 4, 5, 12, 13, 20, and 21, rejected under 35 U.S.C. § 103(a) as unpatentable over Borchardt in view of Mann and further in view of Alexander.

Upon consideration of the record before us, we reverse. We begin with the rejection of claims 1-3, 6-11,14-19, and 22-24

under 35 U.S.C. § 103(a) as unpatentable over Borchardt in view of Mann.

In rejecting claims under 35 U.S.C. § 103, it is incumbent upon the examiner to establish a factual basis to support the legal conclusion of obviousness. See In re Fine, 837 F.2d 1071, 1073, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988). In so doing, the examiner is expected to make the factual determinations set forth in Graham v. John Deere Co., 383 U.S. 1, 17, 148 USPQ 459, 467 (1966), and to provide a reason why one having ordinary skill in the pertinent art would have been led to modify the prior art or to combine prior art references to arrive at the claimed invention. Such reason must stem from some teaching, suggestion or implication in the prior art as a whole or knowledge generally available to one having ordinary skill in the art. Uniroyal, Inc. v. Rudkin-Wiley Corp., 837 F.2d 1044, 1051, 5 USPQ2d 1434, 1438 (Fed. Cir. 1988); Ashland Oil, Inc. v. Delta Resins & Refractories, Inc., 776 F.2d 281, 293, 227 USPQ 657, 664 (Fed. Cir. 1985); ACS Hosp. Sys., Inc. v. Montefiore Hosp., 732 F.2d 1572, 1577, 221 USPQ 929, 933 (Fed. Cir. 1984). These showings by the examiner are an essential part of complying with the burden of presenting a prima facie case of obviousness. Note In re Oetiker, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir.

1992). If that burden is met, the burden then shifts to the applicant to overcome the prima facie case with argument and/or evidence. Obviousness is then determined on the basis of the evidence as a whole. See id.; In re Hedges, 783 F.2d 1038, 1039, 228 USPQ 685, 686 (Fed. Cir. 1986); In re Piasecki, 745 F.2d 1468, 1472, 223 USPQ 785, 788 (Fed. Cir. 1984); and In re Rinehart, 531 F.2d 1048, 1052, 189 USPQ 143, 147 (CCPA 1976).

Turning to claim 1, the examiner asserts (final rejection, page 3) that "Borchardt does not specifically states [sic] enabling said Execution Trace Facility to restart said program and detect and correct said malfunction only in response to a determining [sic] that a malfunction of said program has occurred." The examiner's position (id.) is that "[i]t is well known in the art that when a system to restart the program and detect and correct malfunction, it common [sic] indicates a malfunction has occurred, or when a malfunction has occurred, the system would restart the program and detect and correct the malfunction." The examiner adds (id.) essentially that an artisan would have used this common procedure by enabling the Execution Trace Facility to restart the program and detect and correct the malfunction only in response to determining that a

malfunction of the program has occurred, for better detection and correction of the malfunction.

The examiner additionally asserts (id.) that "Borchardt does not specifically teach inserting said Execution Trace Facility into the code of the program in a disable mode." To overcome this deficiency of Borchardt, the examiner turns to Mann for a teaching of a software debug facility which is able to set the tracing information in an enable or disable mode. The examiner asserts (id.) that "[i]t would have been obvious to one of ordinary skill in the art at the time the invention was made to allow Borchardt's program to be set to a disable mode as taught by Mann because the system will not be interrupted at the disable mode." The examiner adds (answer, page 3) that one of ordinary skill in the art would have been motivated to include Mann's disable function in Borchardt's trace facility in view of Mann's disclosure of the advantage of disabling the trace function to reduce process power consumption.

Appellants assert (brief, page 5) that in Borchardt, the trace program is placed within the program and is executed each time the program is executed, and that in Borchardt, the only decision is whether or not the trace results should be stored in memory or not (discarded). Appellants argue (id.) that the

amended claims expressly recite that the Execution Trace Facility is inserted into the code of the program in a 'disabled mode' and thereafter enabled 'only' in response to a determination that a malfunction of the program has occurred." Appellants assert (id.) that Borchardt "fails to show or suggest in any way the inclusion of an Execution Trace Facility within a program in a disabled mode which may then be selectively enabled upon a determination that an error has occurred." In addition, appellants (brief, pages 5 and 6) dispute the examiner's assertion that Mann discloses a software debug facility, arguing that Mann discloses a debug controlled software program 112 which is executed by host system 111, which controls the extraction and analysis of debug information generated by the target system 101, and that Mann teaches a hardware system which must be coupled to the system under test via a debug port to execute a test of the software within the target system.

It is further argued (brief, page 6) that "no combination of Mann and Borchardt, et al., is appropriate, and even if such a combination is appropriate, that combination cannot be said to show or suggest the insertion of an Execution Trace Facility in a 'disabled' mode 'into the code of the program ...' which is thereafter recompiled and operated 'in a normal operating

environment ...' until such time as a malfunction has occurred. Thereafter, the Execution Trace Facility is enabled and the program is restarted in order to correct that malfunction in accordance with the express recitation within the present claims."

The examiner responds (answer, page 3) by maintaining that Mann discloses a software debug facility in view of the disclosures of Mann (col. 2, lines 59 and 60) that "[w]hat is needed is a software debug system and operating procedure that includes an improved trace capability," and (col. 6, lines 57-59) that "[t]he disabling of trace gathering is advantageously a software option."

Appellants respond (reply brief, page 2) by acknowledging that the software program 112 of Mann is software based, but maintain that "as the software program does not operate within the target system Mann cannot be characterized as a 'software debug facility' as that term is recognized by those having ordinary skill in this art." Appellants additionally argue (id.) that Mann "cannot be said to show or suggest the insertion of a disabled test facility into a software program for selective subsequent enablement in response to detection of a malfunction as set forth within the present claims."

As stated by the court in In re Hiniker Co., 150 F.3d 1362, 1369, 47 USPQ2d 1523, 1529 (Fed. Cir. 1998) "[[t]he name of the game is the claim." Claims will be given their broadest reasonable interpretation consistent with the specification, and limitations appearing in the specification are not to be read into the claims. In re Etter, 756 F.2d 852, 858, 225 USPQ 1, 5 (Fed. cir. 1985). Claim 1 recites "means for inserting said Execution Trace Facility into the code of the program in a disabled mode, means for recompiling said program after said Execute Trace Facility is inserted into said code of the said program, and means for enabling said Execution Trace Facility to restart said program and detect and correct said malfunction only in response to a detection that a malfunction of said program has occurred."

From our review of Borchardt, we find that when a programmer debugs a program, the programmer uses a test scenario designed to force as many as possible logic paths. After inputting the test scenario, the programmer decides if use of the trace facility is desired. If desired, the trace facility becomes active. The initial trace facility options (filtering criteria) are input at this time by the programmer. Upon setting the criteria the trace facility is ready for implementation (col. 3, lines 33-67). The

program is executed while the trace facility is active. The trace facility gathers all historical trace data which may be pertinent. Each trace output entry which is produced during execution is associated with one or more categories of categories/filter of the trace criteria. The trace entries are preferably stored as object oriented objects (col. 4, lines 1-17).

From the disclosure of Borchardt (col. 3, lines 46 and 47) that "[i]f desired, the trace facility of the present invention becomes active," we find that the trace facility was inactive (disabled) prior to becoming active by input of data by the programmer (or through default criteria), as is clear from figure 2. Although both the examiner (final rejection, page 3) and appellants (brief, page 5) take the position that Borchardt does not disclose inserting the Execution Trace Facility into the program code in a disabled mode, we find that because the trace facility of Borchardt is inactive until activated by the programmer, that the trace facility was inherently inserted into the program code in a disabled mode.

Moreover, claim 1 additionally requires that the program is recompiled after the Execution Trace Facility has been inserted into the program code. Borchardt discloses (col. 2, lines 9-12)

that "an object of the present invention is to provide a system and method for permitting a programmer to obtain and re-filter historical trace information without re-executing the program being debugged." The examiner's position (final rejection, page 2) is that figure 2 of Borchardt discloses means for compiling the program after the Execution Trace Facility has been inserted into the code of the program. From our review of Borchardt, and in particular figure 2, we find that Borchardt discloses executing the program after the filtering criteria have been set by the program or by default, but find no teaching or suggestion of recompiling the program after insertion of the Execution Trace Facility. Executing a program and recompiling a program are not the same. In addition, we find no teaching or suggestion in Borchardt of restarting the program and detecting and correcting malfunctions only in response to a detection that a malfunction has occurred in the program. We are not persuaded by the examiner's assertion (final rejection, page 3) to the effect that it is well known to restart a program and detect and correct malfunctions after malfunctions have occurred, and that for better detecting and correcting of malfunctions, an artisan would have restarted the program and detected and corrected the malfunction only after a malfunction is detected. The

examiner's's opinion is unsupported in the record, and is not a substitute for evidence. Thus, we find that Borchardt does not suggest enabling the Execution Trace Facility only in response to a determination that a malfunction has occurred. We are cognizant of the disclosure of Borchardt that the programmer debugs the program (col. 3, lines 7 and 38). However, we find no teaching or suggestion in Borchardt, and none has been pointed to by the examiner, that would have suggested enabling the Execution Trace Facility only in response to a determination that a malfunction has occurred.

Turning to Mann, from our review of Mann, we find that trace control circuit 218 implements user control for selectively activating and deactivating trace functionality (col. 6, lines 17-24). We agree with the examiner that Mann discloses enabling and disabling the tracing function. In addition, from the disclosure of Mann (col. 6,, lines 57-59) that the disabling of trace gathering is advantageously a software option, reducing process power consumption and eliminating natural throttle-back tendencies, we find that an artisan would have been motivated to enable and disable the trace function of Borchardt. However, claim 1 requires more. The examiner's assertion (final rejection, page 3) that Mann teaches enabling and disabling the

tracing function ignores the portions for the claim requiring that after recompiling the program, enabling the Execution Trace Facility to restart the program and detect and correct the malfunction only in response to detection of a malfunction. In addition, because Borchardt discloses that the programmer makes the determination as to whether the trace facility should be activated, we find no suggestion or motivation to enable the Execution Trace Facility to be enabled only in response to detection of a malfunction. Thus, even if we combined the teachings of Borchardt and Mann, the references are silent as to recompiling the program after insertion of the Execution Trace Facility, and enabling the Execution trace facility to restart the program and to detect and correct a malfunction only in response to a detection that a malfunction of the program has occurred.

From all of the above, we find that the examiner has failed to establish a prima facie case of obviousness of claim 1. The rejection of claim 1 under 35 U.S.C. § 103(a), and claims 2, 3, and 6-8, dependent therefrom, is therefore reversed. As independent claims 9 and 17 include similar limitations with respect to recompiling the program after the Execution Trace Facility has been inserted into the program code, and restarting

the program and detecting and correcting the malfunction only in response to a determination that a malfunction has occurred, the rejection of independent claims 9 and 17, as well as claims 10, 11, 14-18, and 22-24, dependent therefrom, is reversed.

We turn next to the rejection of claims 4, 5, 12, 13, 20, and 21 under 35 U.S.C. § 103(a) as unpatentable over Borchardt in view of Mann, and further in view of Alexander. We reverse the rejection of claim 4, 5, 12, 13, 20, and 21 under 35 U.S.C. § 103(a) because Alexander does not make up for the basic deficiencies of Borchardt and Mann.

CONCLUSION

To summarize, the decision of the examiner to reject claims 1-24 under 35 U.S.C. § 103 is reversed.

REVERSED

MICHAEL R. FLEMING Administrative Patent Judge	)
STUART S. LEVY Administrative Patent Judge	) BOARD OF PATENT APPEALS AND INTERFERENCES
HOWARD B. BLANKENSHIP Administrative Patent Judge	)

SSL/kis

BRACEWELL & PATTERSON, LLP
INTELLECTUAL PROPERTY LAW
P. O. BOX 969
AUSTIN, TX 78767-0969